



Detecting HI Galaxies with Deep Neural Networks in the Presence of Radio Frequency Interference

Ruxi Liang^{1,2} , Furen Deng^{1,2} , Zepei Yang³, Chunming Li⁴, Feiyu Zhao^{2,5}, Botao Yang⁴, Shuanghao Shu^{1,2}, Wenxiu Yang^{1,2}, Shifan Zuo^{1,2,7}, Yichao Li⁶ , Yougang Wang^{1,2,6,7}, and Xuelei Chen^{1,2,6,7}

¹ National Astronomical Observatories, Chinese Academy of Sciences, Beijing 100101, China; wangyg@bao.ac.cn, xuelei@bao.ac.cn

² University of Chinese Academy of Sciences, Beijing 101408, China

³ Department of Physics, Northeastern University, Boston, MA 02115, USA

⁴ Biomedical Instrument Institute, School of Biomedical Engineering, Shanghai Jiao Tong University, Shanghai 200030, China

⁵ Shanghai Astronomical Observatory, Chinese Academy of Sciences, Shanghai 200030, China

⁶ Key Laboratory of Cosmology and Astrophysics (Liaoning), College of Sciences, Northeastern University, Shenyang 110819, China

⁷ Key Laboratory of Radio Astronomy and Technology, Chinese Academy of Sciences, Beijing 100101, China

Received 2023 April 9; revised 2023 April 19; accepted 2023 April 24; published 2023 October 4

Abstract

In the neutral hydrogen (HI) galaxy survey, a significant challenge is to identify and extract the HI galaxy signal from the observational data contaminated by radio frequency interference (RFI). For a drift-scan survey, or more generally a survey of a spatially continuous region, in the time-ordered spectral data, the HI galaxies and RFI all appear as regions that extend an area in the time-frequency waterfall plot, so the extraction of the HI galaxies and RFI from such data can be regarded as an image segmentation problem, and machine-learning methods can be applied to solve such problems. In this study, we develop a method to effectively detect and extract signals of HI galaxies based on a Mask R-CNN network combined with the PointRend method. By simulating FAST-observed galaxy signals and potential RFI impact, we created a realistic data set for the training and testing of our neural network. We compared five different architectures and selected the best-performing one. This architecture successfully performs instance segmentation of HI galaxy signals in the RFI-contaminated time-ordered data, achieving a precision of 98.64% and a recall of 93.59%.

Key words: methods: data analysis – methods: observational – techniques: image processing

1. Introduction

In recent years, a number of advanced radio telescopes and arrays have been constructed, including the Five-hundred-meter Aperture Spherical radio Telescope (FAST; Nan et al. 2011), the Australian Square Kilometre Array Pathfinder (ASKAP; Johnston et al. 2008), and MeerKat (Booth & Jonas 2012), among others. In the coming decade, the next generation of radio telescope arrays, such as the Square Kilometre Array (SKA; Dewdney et al. 2009), are anticipated to be completed. The study of neutral hydrogen is one of the primary scientific goals of these telescopes, and HI galaxy surveys are key observations of them (Tolley et al. 2022). From the HI galaxy survey data, we can examine the HI content and mass function of the galaxies, gas accretion, the correlation between HI and star formation, and the influence of the environment on HI (Giovanelli & Haynes 2015). These sensitive and precise instruments demand more sophisticated observational techniques and signal-processing methods.

The HI Parkes All-Sky Survey (HIPASS; Meyer et al. 2004) and the Arecibo Legacy Fast ALFA Survey (ALFALFA; Giovanelli et al. 2005; Jones et al. 2018) are the most extensive HI surveys completed so far. The “multifind” peak-finding

algorithm (Kilborn 2001) was employed to identify and filter data signal peaks in the HIPASS data processing. This method searches local maxima in data cubes and identifies potential signals by setting a threshold. The ALFALFA survey used a matched filtering algorithm (Saintonge 2007), which is sensitive to wide and weak signals. Although these algorithms have served these surveys successfully, they still exhibit some shortcomings. The multifind result is sensitive to the threshold, and has difficulty with overlapping signals, or signals with unusual shapes and features. The matched filtering algorithm also relies on assumptions about signal shapes, necessitating adjustments to algorithm parameters based on extensive experimentation and experience, it is prone to false alarms and missed detections when encountering multiple local maxima. The identification of radio frequency interference (RFI) is also far from perfect for these algorithms. More advanced and robust signal extraction methods are needed for future HI surveys.

The RFI is always a challenge that radio astronomical observations face. RFI sources can be artificial or natural, with the former including digital television, mobile and satellite communications, and so on (Fridman & Baan 2001). Efficient

RFI mitigation algorithms that can identify the RFI are essential for radio astronomical observations. Many automatic RFI flagging algorithms have been developed, typically by looking for unusually large deviations in the sample. For example, the widely used Sum-Threshold method (Offringa et al. 2010) searches the RFI of different possible time and frequency spread by scanning the data with a sliding window, and comparing the sum of the power of consecutive samples with a blocksize-dependent threshold.

RFI mitigation and celestial signal extraction are two sides of the same process. In past and present HI observations, the usual practice is to first remove the various interferences, including standing waves and RFI, through a pipeline, and then extract the desired HI signal from the processed data. However, the identification of RFI is not absolute, and the extraction process still faces the influence of some interference. Moreover, RFI often superimposes HI signals, causing contamination and rendering the data unusable. Therefore, a major challenge in the data processing of HI galaxy surveys is to identify the extragalactic HI signals amidst a vast amount of data.

In recent years, there has been much advancement in machine learning (ML), and it has been applied to various research directions in astronomy (Ball & Brunner 2010). In particular, these techniques have been applied to radio astronomical data processing tasks, such as RFI identification and mitigation, celestial source detection and classification, and the analysis of observational data, among others (Baron 2019). Numerous deep-learning-based models have been applied to identify and mitigate RFI, especially the Convolutional Neural Networks (CNN) (Pinchuk & Margot 2022; Sun et al. 2022), U-Net (Akeret et al. 2017a; Yang et al. 2020), and so on. Other ML-based image-processing models have also been applied to astronomy, such as the new source finder developed by Riggi et al. (2023) based on the Mask R-CNN framework for detecting and classifying sources in radio continuum images.

Mask R-CNN is a CNN-based object detection and instance segmentation framework, which has achieved remarkable results in the field of computer vision (He et al. 2017). PointRend is a technique for improving image segmentation results by adding a rendering approach on top of the existing network, presenting fine-grained object boundaries through adaptive point sampling and label estimation (Kirillov et al. 2020). This technique enhances segmentation quality, producing more refined edges.

Inspired by these works, we apply the Mask R-CNN model and PointRend method to HI signal extraction in radio telescope data processing, hoping to more accurately detect and segment target objects in astronomical images. We develop an HI galaxy-searching method based on the Mask R-CNN model and the PointRend method. The model can directly search for and identify HI galaxies in time-order data contaminated by RFI, and can extract signals by segmenting

the data. Using FAST as an example, we simulated the observed HI galaxies and potential RFI impacts, and then trained, refined, and selected different architectures of PointRend Mask R-CNN models, ultimately achieving a good performance in identifying galaxy signals.

The structure of this paper is as follows. Section 2 of the paper introduces the machine-learning methods we used, including the principles of Mask R-CNN and PointRend, the network structure we employed, and the model evaluation method. In Section 3, the data preparation process is expounded. Section 4 presents the training and testing of the networks, while Section 5 presents the final results of our experiment. Section 6 provides further analysis and discussion of the results. Finally, Section 7 summarizes the entire paper.

2. Method

2.1. Machine Learning Method: Mask R-CNN and PointRend

In this study, we developed the Mask R-CNN network by integrating it with the PointRend method to accomplish the instance segmentation task of identifying HI galaxies in astronomical observation data.

Mask R-CNN is an improved version of Faster R-CNN, which is a classic two-stage object detection network. The Faster R-CNN represents detected objects by generating bounding boxes and corresponding class information (Ren et al. 2015). Mask R-CNN adds a mask branch to the Faster R-CNN network, which could generate the binary mask for each detected object. The additional mask branch significantly improves the network performance in instance segmentation tasks.

The PointRend method is an innovative strategy that can be integrated with various neural networks. By adding a PointRend head to the network, it improves the accuracy and resolution of image segmentation (Kirillov et al. 2020). After obtaining a preliminary coarse mask through other networks, PointRend generates some sampling points concentrated in the areas where the segmentation results are uncertain, then adopts a sub-network called PointRend head to predict the classification of these points based on the input feature map. The predicted information is then combined with the coarse mask to generate a more precise mask.

Figure 1 shows the network structure of the model we used. For single-channel two-dimensional data, the model first obtains a feature map through the backbone network, which serves as the input for the Region Proposal Network (RPN; Ren et al. 2015) and the final PointRend Head. The RPN generates a series of proposals from the input feature map, each with a specific region where the target object might be located. Each proposal is combined with the feature map to generate a series of RoIs (Regions of Interest), which are then passed through the RoI Align Layer (He et al. 2017) to obtain fixed-size feature

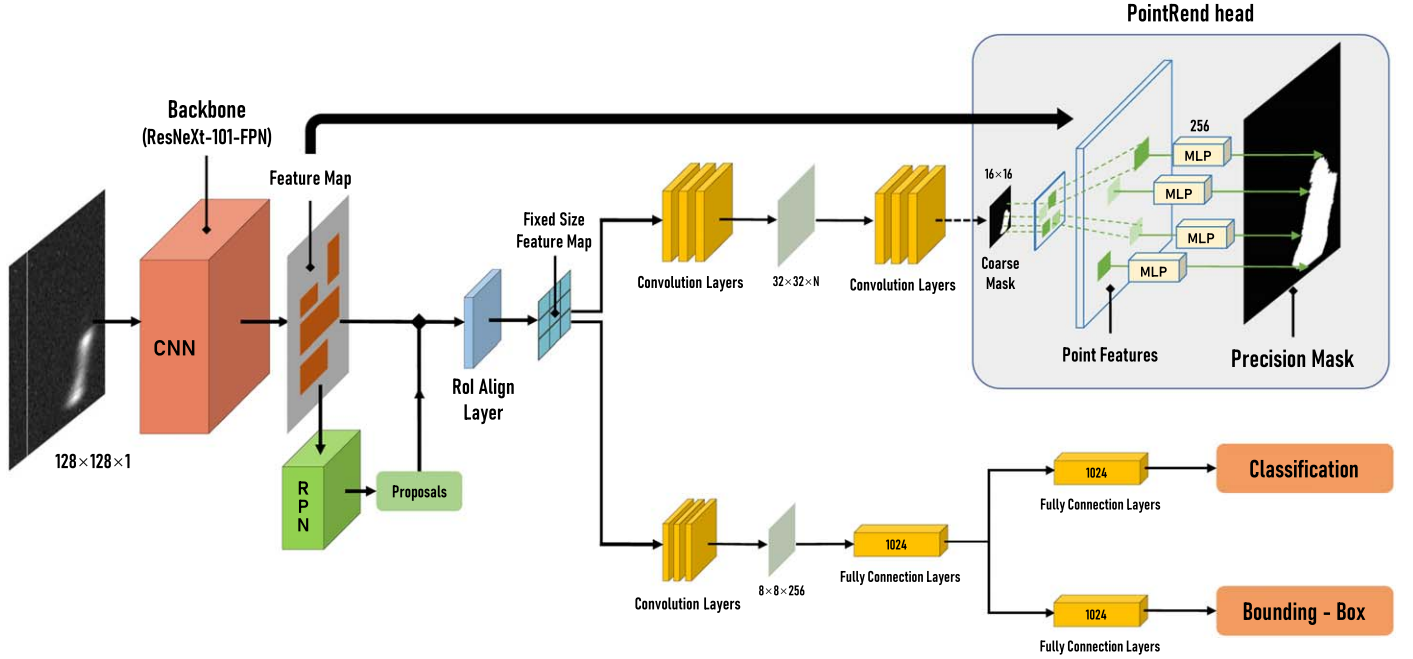


Figure 1. Schematic diagram of the PointRender Mask R-CNN model architecture.

maps of uniform size. There are two branches next. The branch at the bottom of the structure diagram is the RoI head branch, which has the same structure as the corresponding part of Faster R-CNN. It first transforms the fixed-size feature map into a series of smaller maps by going through convolutional layers, and then obtains the class and bounding box information through several fully connected layers. This branch essentially completes the object detection of the input data. The upper branch is the Mask branch. After passing the fixed-size feature map through a series of convolutional layers, we obtain a coarse mask.

Next is the PointRender part of the network. PointRender uses a sampling strategy based on uncertainty to generate some sampling points for refinement according to the coarse mask information. The model employs an additional sub-network called the PointRender Head, which receives the selected refinement points and the high-resolution feature map generated at the beginning of the entire network as input, and predicts the classification of each sampling point through a series of MLPs (Multi-Layer Perception). Finally, the predicted class information is combined with the coarse mask to obtain a more accurate final precision mask. In each iteration, the sub-network calculates the uncertainty of the class prediction for each point, and selects a certain portion of points for updating based on the uncertainty values. This means the selected points are mainly located in the detailed areas of the segmentation result (i.e., the edge areas and texture-complex areas), which are the areas that need improvement the most in the segmentation results.

By undergoing these processes for each RoI, we can complete the instance segmentation of all targets in the input data. All parts of the model participate in training. By minimizing the loss function, the model updates the parameters of each network through backpropagation. We define the multi-task loss on each sampled RoI as

$$L_{\text{total}} = L_{\text{cls}} + L_{\text{box}} + L_{\text{mask}} + L_{\text{PointRender}}. \quad (1)$$

L_{cls} is the classification loss defined as

$$L_{\text{cls}} = -\log(p_u), \quad (2)$$

where p_u is the probability of an RoI belonging to the true class label u ($u \in \{1, 2, \dots, C\}$), calculated by the softmax function:

$$p_u = \frac{\exp(z_u)}{\sum_{k=1}^C \exp(z_k)},$$

where z_i represents the score of the RoI belonging to the i -th class for $i \in \{1, 2, \dots, C\}$. L_{box} is the bounding-box regression loss, which describes the bounding-box branch's ability to localize objects during bounding-box regression, defined as:

$$L_{\text{box}} = \sum_{i \in \{x, y, w, h\}} \text{Smooth}_{L_1}(t_i^u - v_i), \quad (3)$$

where $v = (v_x, v_y, v_w, v_h)$ is the true bounding-box regression target for class u and $t^u = (t_x^u, t_y^u, t_w^u, t_h^u)$ is the predicted bounding box coordinate. Smooth_{L_1} represents the Smooth L_1

loss function:

$$\text{Smooth}_{L_1} = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases}. \quad (4)$$

L_{mask} is described as the average binary cross-entropy per pixel and describes the mask head's ability to classify each pixel, defined as

$$L_{\text{mask}} = -\frac{1}{m^2} \sum_{i=1}^{m^2} [p_i \log(\hat{p}_i) + (1 - p_i) \log(1 - \hat{p}_i)], \quad (5)$$

where m^2 is the total number of pixels in the mask (16×16 in our network), p_i is the true value of the i -th pixel, and \hat{p}_i is the predicted value of the i -th pixel.

The PointRend loss, $L_{\text{PointRend}}$, calculates binary cross-entropy only on the sampled points that need to be refined:

$$L_{\text{PointRend}} = -\frac{1}{N_{\text{point}}} \sum_{i=1}^{N_{\text{point}}} [p_i \log(\hat{p}_i) + (1 - p_i) \log(1 - \hat{p}_i)], \quad (6)$$

where N_{point} represents the number of sampled points that need refinement (set to 10 in our task), p_i is the true value of the i -th point, and \hat{p}_i is the predicted value of the i -th point. In addition, the RPN network used in the model has its own loss function and is trained independently during the training process.

2.2. Model Evaluation

In our PointRend Mask R-CNN network, we selected five distinct backbones for obtaining feature maps and conducted a comparative analysis to evaluate the impact of different backbones on the network performance. Our choices are all based on Residual Networks (ResNet; He et al. 2016), which is a deep convolutional neural network. The ResNet introduces residual connections to solve the problem of gradient vanishing and explosion issues during the training of deep neural networks.

In our selection, ResNet-50-FPN and ResNet-101-FPN are Feature Pyramid Networks (FPN; Lin et al. 2017) based on 50-layer and 101-layer residual networks, respectively. These FPNs add a top-down pathway and lateral connections to the original ResNet, enabling the network to better capture features at different scales, and could improve object detection and instance segmentation performance by leveraging multi-scale features. ResNet-50-C5-Dilated and ResNet-101-C5-Dilated are dilated convolutional networks based on 50-layer and 101-layer residual networks, respectively, using dilated convolution in the last convolutional layer (C5) (Yu & Koltun 2016). This approach increases the receptive field size, thereby improving the detection and segmentation performance for large-scale objects. Our primary focus is on the ResNeXt-101-FPN backbone. ResNeXt-101 is an improved 101-layer ResNet network that employs grouped convolution on top of ResNet,

dividing the input channels into multiple groups and performing convolution operations within each group. This enhances the network's expressiveness and parameter efficiency, allowing for improved performance with relatively low computational complexity (Xie et al. 2017). After combining with FPN, ResNeXt-101-FPN should perform slightly better than ResNet-101-FPN theoretically.

Generally, deeper network structures can usually learn more diverse feature representations, thereby improving the accuracy of instance segmentation. FPN can effectively capture multi-scale feature information by integrating features from different levels, resulting in better performance when dealing with objects of varying sizes. Dilated convolution, by expanding the receptive field of the convolution kernel, can better capture information from large-scale objects. For our project, an FPN with a deeper structure is theoretically more suitable.

We employed the precision, recall rate, and F1 score, which are commonly used performance metrics for evaluating image segmentation models (Forsyth & Ponce 2002), to evaluate our method. The precision in this case is the proportion of true galaxies among the samples classified as galaxies by the model, reflecting the accuracy of the model in recognizing galaxies.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad (7)$$

where TP represents a correct segmentation (detection) that the instance is classified as a member of the class while FP represents an incorrect segmentation of such classification. Precision belongs to $[0, 1]$, and a higher value indicates that the model is less likely to misidentify.

The recall rate in the present case is the fraction of identified H I galaxies among all H I galaxies, representing the model's capability of detection. Recall belongs to $[0, 1]$, and a higher value indicates a stronger recognition ability. It is defined as

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad (8)$$

where FN represents an incorrect segmentation that the instance is not classified as a member of the class.

The F1 score is the harmonic mean of the precision and the recall, also belonging to $[0, 1]$, providing a comprehensive evaluation of both precision and recall performance. It can serve as the standard for assessing the model, and a higher value indicates the model has a better performance. It is defined as

$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (9)$$

3. Mock Data

The PointRend Mask R-CNN is a supervised neural network model that requires data for training and testing. For our mission, the construction of data sets can be diverse. Referring

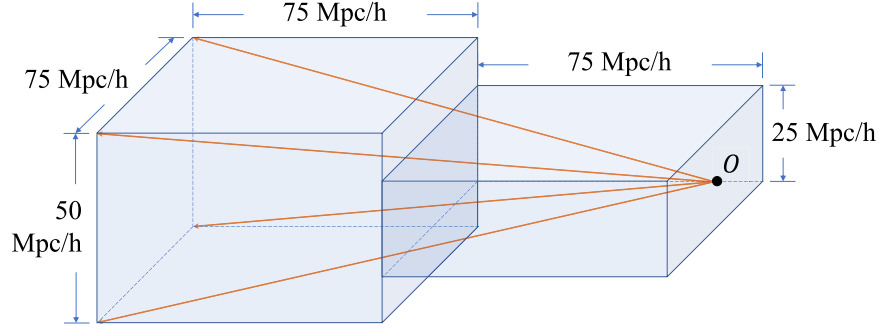


Figure 2. The configuration of the light cone by stacking two boxes. The size of two boxes is shown in the figure, O is the observer and the orange lines join the observer with the four corners of the field of view. The redshift range of the light cone extends to $z \approx 0.05$, and its angular area is approximately $28 \times 19 \text{ deg}^2$.

to the FAST telescope, we simulated the HI galaxy signals it could observe and the possible RFI effects it might encounter.

3.1. HI Galaxies Data Simulation

We generate the mock HI galaxies from the IllustrisTNG magnetohydrodynamical (Weinberger et al. 2017; Pillepich et al. 2018) simulation. It includes physical processes such as gas cooling, star formation, stellar evolution, metal enrichment, black hole growth, stellar winds, supernovae, and active galactic nuclei (AGNs), and is consistent with current observations. In this work, we use the TNG100-1 data set. The box size for TNG100 is 75 Mpc/h and mass resolution is $9.4 \times 10^5 M_\odot/h$ for the baryon particle and $5.1 \times 10^6 M_\odot/h$ for the dark matter particle. The box size ensures that there are enough galaxies as training sets and test sets, and the mass resolution ensures that the structure of galaxies can be resolved. The IllustrisTNG adopted the Planck 2015 cosmological parameters (Ade et al. 2016), i.e., $\Omega_\Lambda = 0.6911$, $\Omega_m = 0.3089$, $\Omega_b = 0.0486$, $\sigma_8 = 0.8159$, $n_s = 0.9667$, and $h = 0.6774$, and we adopt this model throughout the paper.

The total mass of atomic and molecular hydrogen for each gas particle can be obtained directly from the TNG100 catalog. However, these two parts are not separated in the simulation. Diemer et al. (2018) has separated the molecular and atomic hydrogen contents for galaxies in TNG100. However, we also need to take into account the velocity of each gas particle to get the spectral profile for each galaxy, which cannot be obtained from the existing catalog. We calculate the HI mass for each gas particle based on the method of Gnedin & Kravtsov (2011). We refer readers to Deng et al. (2022) for details of the calculation. Following Diemer et al. (2018) we consider only galaxies with stellar mass or gas mass greater than $2 \times 10^8 M_\odot$, which are well represented by particles in this simulation.

We assume that the properties of HI galaxies do not evolve significantly over the small redshift range considered in this work, and only use the simulation snapshot at $z \approx 0$. The box is split into two boxes with size $75 \times 75 \times 50 (\text{Mpc}/h)^3$ and $75 \times 75 \times 25 (\text{Mpc}/h)^3$. Then we stack the two boxes to form a

light cone volume as shown in Figure 2, where O is the observer. We have a rectangular field of view and the orange lines join the observer with the four corners of the field. We choose this configuration to ensure sufficient redshift coverage and field of view while avoiding the repeating of galaxy samples. The redshift range of the light cone extends to $z \approx 0.05$, and its angular area is approximately $28 \times 19 \text{ deg}^2$. We then deposit the gas particles into angular and frequency grids, where the angular grid has a size of $\Delta\theta = 0''.5$, well below the beam resolution of FAST, and the frequency grid has a size of $\Delta\nu = 0.02 \text{ MHz}$, to suit the purpose of the galaxy detection. The frequency of each gas particle is determined as $\nu = \nu_{21}/(1+z)/(1+\beta)$, where $\nu_{21} \approx 1420.4 \text{ MHz}$ is the rest-frame frequency of 21 cm radiation, z is the cosmological redshift, and β is the line-of-sight component of peculiar velocity in units of the speed of light.

We calculate the brightness temperature for cell i in frequency ν by

$$T_b^i(\nu) = \frac{3c^2}{32\pi\nu_{21}^3} A_{10} \frac{h\nu^2}{k_B m_p} \frac{\Delta M_{\text{HI}}^i}{D_A(z)^2 \Delta\theta^2 \Delta\nu}, \quad (10)$$

where c is the speed of light, $A_{10} \approx 2.85 \times 10^{-15} \text{ Hz}$ is the spontaneous emission coefficient of the 21 cm transition, m_p is the mass of the proton, k_B is the Boltzmann constant, $D_A(z)$ is the angular diameter distance, and ΔM_{HI}^i is the HI mass in cell i . We ignored the velocity dispersion inside the gas particle in our calculation, which may smooth the spectrum but cannot be obtained from the simulation. The spectrum of one simulated galaxy is shown in Figure 3. Its peak flux is about 8 mJy and the line width is about 1 MHz with a characteristic “double horn” profile. It is consistent with our knowledge about the HI profile in low redshift galaxies (Saintonge 2007).

We model the beam of the FAST as a Gaussian function with $\sigma = 0.518\lambda/(300 \text{ m})$, though the real beam may have a more complicated dependence on the frequency. The 19 beams are rotated 23.4° w.r.t the configuration given in Jiang et al. (2020), to achieve a more uniform coverage in the drift scan. We place the angular center of the grids at the zenith, and assume the sky

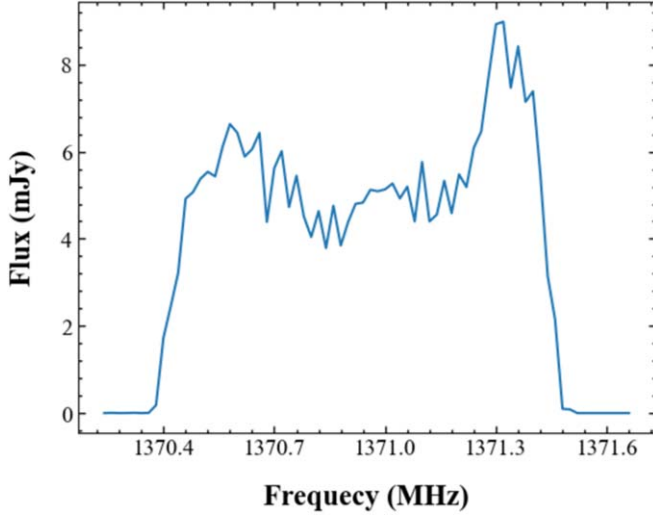


Figure 3. The spectrum of one of our simulated galaxies. The peak flux is about 8 mJy and the line width is about 1 MHz with a characteristic “double horn” profile.

is surveyed with the rotation of the Earth. The scan produces strips along the R.A. (right ascension) direction. According to Jiang et al. (2020), the 19 beams cover over $25'$ in the direction of decl. (declinations), so by repeatedly scanning different declinations with a separation of $10'$, the whole available angular area (which is approximately $28 \times 19 \text{ deg}^2$) is surveyed. We set the time resolution as 1.00663296 s and frequency resolution as 0.02 MHz.

Based on the sensitivity of the FAST, we dropped the galaxies with fluxes below 5 mJy. Additionally, when training with data from a single beam, we also removed those galaxies that could be observed by other beams but were not visible to the specific beam we use. Ultimately, we obtained 4495 HI galaxies. These galaxies include both bright and faint ones, with varying shapes, and produce different brightness levels in the time order data, essentially covering various scenarios encountered in real observations. Figure 4 illustrates a piece of simulated HI galaxy signals, containing only galaxy signals without RFI, allowing us to label each galaxy easily and conveniently.

3.2. RFI Simulation

There are a number of software packages that can be employed for simulating RFI. The `HIDE` (HI Data Emulator) is a software package for simulating HI observation data, and it could also generate mock RFI (Akeret et al. 2017b; Yang et al. 2020). The `Hera_sim` (Kerrigan et al. 2019) is a Python software package developed for simulating the Hydrogen Epoch of Reionization Array (HERA) data, which can also generate RFI data (Sun et al. 2022). We integrated and adapted these two software packages to simulate RFI. We considered

several types of RFI, including narrowband RFI, broadband RFI, and “clump” RFI.

Broadband RFI is instantaneous and intense, typically originating from sources such as lightning and transmission cables, generally covering many frequency bands and manifesting as “bright lines” spanning numerous frequencies in time-ordered data. Narrowband RFI is usually caused by digital television, satellite communications, and mobile communication. A typical narrowband RFI appears as a long-lasting and narrow frequency spread signal, presenting as intermittent stripes in time-ordered data. Another type of RFI, with a frequency spread and appearance time similar to galaxies, may stem from harmonics of satellite communications and certain short-term electromagnetic wave emissions. It exhibits a stain-like clump shape in time-ordered data and is more prone to confusion with galaxies. Ultimately, we successfully simulated these types of RFI. We then generated system noise following the method in Jiang et al. (2020) and added it to the data. Figure 5 shows a segment of the mock RFI and noise data, displaying different types of RFI.

4. Model Training and Testing

With the mock data generated above, we trained our network model. We divided the data set into a training set, a validation set, and a test set with a 3:1:1 ratio, then we trained and tested the PointRend Mask R-CNN model with five different backbones.

We set the batch size to 16, and the maximum number of iterations to 50,000. Each training sample generates classes, bounding-boxes, and mask predictions after training. The network also parses the classes and bounding-boxes information from the true mask, which serves as the ground truth. By calculating and minimizing the loss function value according to the method presented in Section 2.1, the model parameters are updated through backpropagation, thereby training the model. We employed the SGDM (Stochastic Gradient Descent with Momentum, Qian 1999) method to update the parameters, which could help accelerate convergence, and set the momentum as 0.9 (Sutskever et al. 2013).

The base learning rate was set as 0.0005. We utilized a learning rate warmup strategy (Goyal et al. 2018), in which the learning rate will increase gradually and linearly from a lower value to the preset base learning rate during the initial stage of training. This strategy helps the model converge more stably in the early training phase and reduces the risk of gradient explosion. We also employed a multi-step learning rate decay strategy (Krizhevsky et al. 2012), decaying the learning rate at the 10,000th and 30,000th iterations by multiplying the preset decay factor (set as 0.6) with the current learning rate. This strategy assists in fine-tuning the model parameters during the later stages of training, providing more precise adjustments as the training progresses, thereby enhancing the model’s

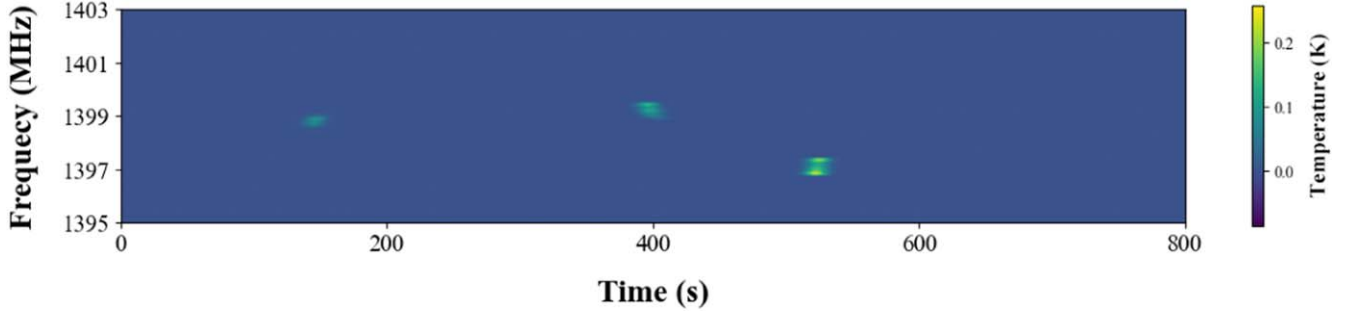


Figure 4. A piece of simulated TOD data of H I galaxies, with the horizontal axis representing time, the vertical axis representing frequency, and the color representing the antenna temperature value. As one can see, there are three galaxies present in the figure.

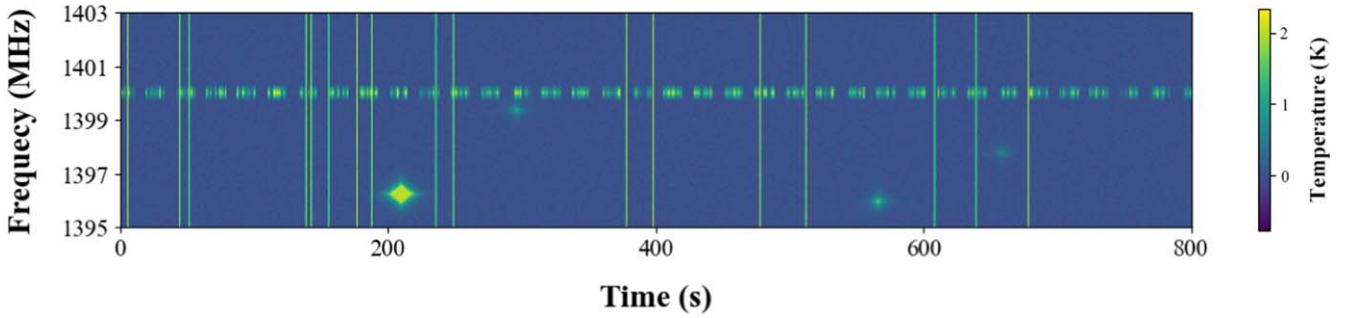


Figure 5. A piece of simulated TOD data of RFI, with the horizontal axis representing time, the vertical axis representing frequency, and the color representing the antenna temperature value. Narrowband RFI manifests as discontinuous horizontal lines with width, broadband RFI appears as thin vertical lines and stain-like RFI presents large and small radiating spots.

performance. Both strategies have been widely used in deep learning. Weight decay was employed as a regularization technique to prevent overfitting and enhance the model's generalization capabilities, with the weight decay coefficient set as 0.0001. This is also a widely applied strategy (Goodfellow et al. 2016).

For the backbone of our main concern, ResNeXt-101-FPN, Figure 6 demonstrates the variations of the loss function values on the training and validation sets during the training process. As one can see, the loss function exhibits a decreasing trend and eventually converges on the training set. On the validation set, the loss function also displays a general decreasing and converging trend. Although slight oscillations in the loss function values on the validation set appear after 12,000 steps and intensify after 32,000 steps, the function values do not have an increasing trend, indicating that our model does not exhibit significant overfitting. Such oscillations are normal and can be attributed to the inherent randomness in the optimization process, and this intensification in the later stages of training may be related to the size of the validation set and the batch size settings. In our mission, if the model experiences underfitting, it may not effectively learn and recognize various galaxy signals within the data, resulting in a low recognition capability. In contrast, in the case of overfitting, the model

could become overly focused on the features within the current training data, leading to a decline in generalization performance on new data. To minimize the occurrence of both underfitting and overfitting, we monitored the model's performance on the validation set and ensured that the oscillations are within an acceptable range. Ultimately, we chose the network trained to 32,000 steps as our model. Similar phenomena were observed in the training processes of the networks with other backbones, and we also selected the final model for each backbone at the training step where the loss function had relatively converged on the training set and before the intensification of oscillations on the validation set.

After training, we tested the model using the test set and calculated the model evaluation metrics according to the method in Section 2.2. The model was trained on NVIDIA GeForce RTX 2080 Ti GPU.

5. Results

For the training results of the PointRender Mask R-CNN model with different backbones, we calculated their precision, recall, and F1 score respectively, as shown in Table 1.

The ResNet-50-C5-Dilated and ResNet-101-C5-Dilated backbones performed well in terms of precision, but their

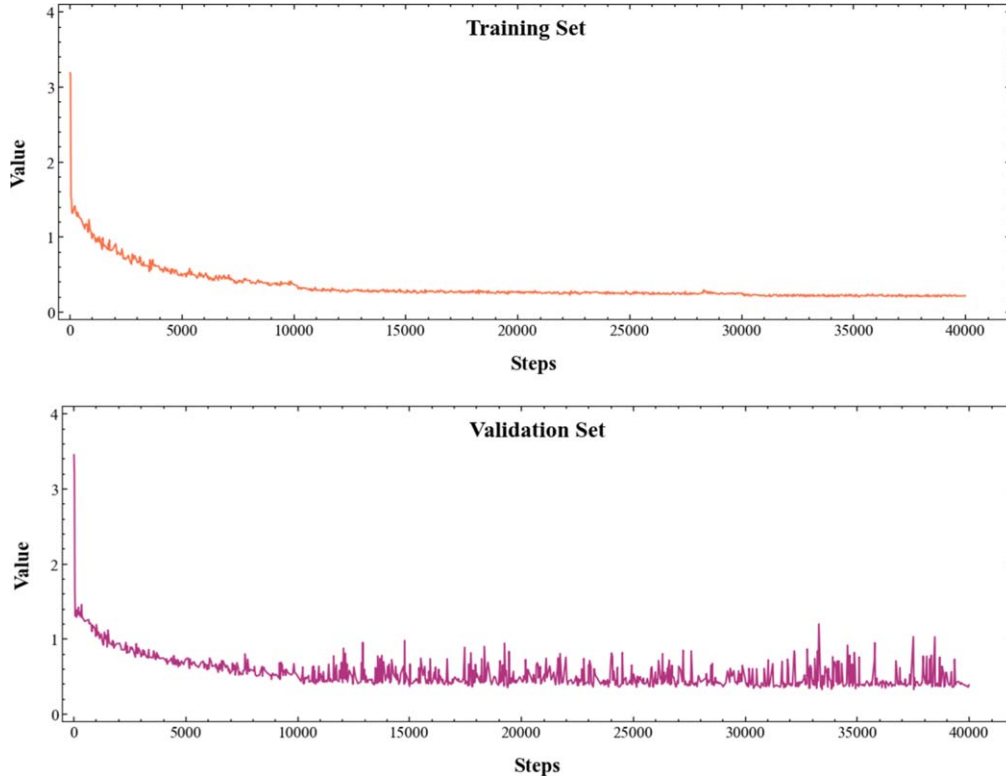


Figure 6. Loss function values of the model with the ResNeXt-101-FPN backbone during the training process. The upper panel illustrates the variation of loss function values on the training set, while the lower panel shows the variation of loss function values on the validation set. The horizontal axis represents the training iteration steps, and the vertical axis indicates the function values. As can be seen, the loss function eventually converges on the training set. On the validation set, the loss function also exhibits a decreasing and converging trend. However, slight oscillations occur after 12,000 steps and intensify after 32,000 steps. Ultimately, we chose the model trained up to 32,000 steps.

Table 1
Precision, recall and F1 score of our PointRend Mask R-CNN network with different backbones.

	Backbone	Precision	Recall	F1 Score
PointRend Mask R-CNN	ResNet-50-FPN	96.15%	94.93%	95.54%
	ResNet-50-C5-Dilated	100%	65.38%	79.07%
	ResNet-101-FPN	92.68%	97.43%	95.00%
	ResNet-101-C5-Dilated	98.14%	67.94%	80.29%
	ResNeXt-101-FPN	98.64%	93.59%	96.05%

recall is quite poor, leading to a low F1 score. The F1 score for the ResNet with FPN is generally better than that of the C5-Dilated ResNet. Moreover, the performance of ResNeXt-101-FPN is slightly better than that of the ResNet-50-FPN and ResNet-101-FPN, which is consistent with our expectations. Ultimately, we selected ResNeXt-101-FPN as the backbone for our model.

Figure 7 illustrates some examples of galaxies correctly recognized by the model, with the yellow lines delineating the regions determined by the model's output mask, and the green lines representing the ground truth. In Figure 7(a), there is a bright RFI spot on the right side, with one galaxy contaminated

by broadband RFI, but the model still accurately identifies all the two galaxies, successfully detects multiple targets. From Figure 7(b), one can see that our model can also effectively discern galaxy data contaminated by narrowband RFI.

When a bright galaxy is encountered, as illustrated in Figure 7(c), our model does not simply identify the “bright” regions, but also captures the faint areas at the edges of the galaxy data, indicating that the model has learned the characteristics of H I galaxies during training. From an image-processing perspective, the high gradient at the edges of such bright galaxies could easily lead to overfitting during training, but our model does not exhibit this issue. Figure 7(d)

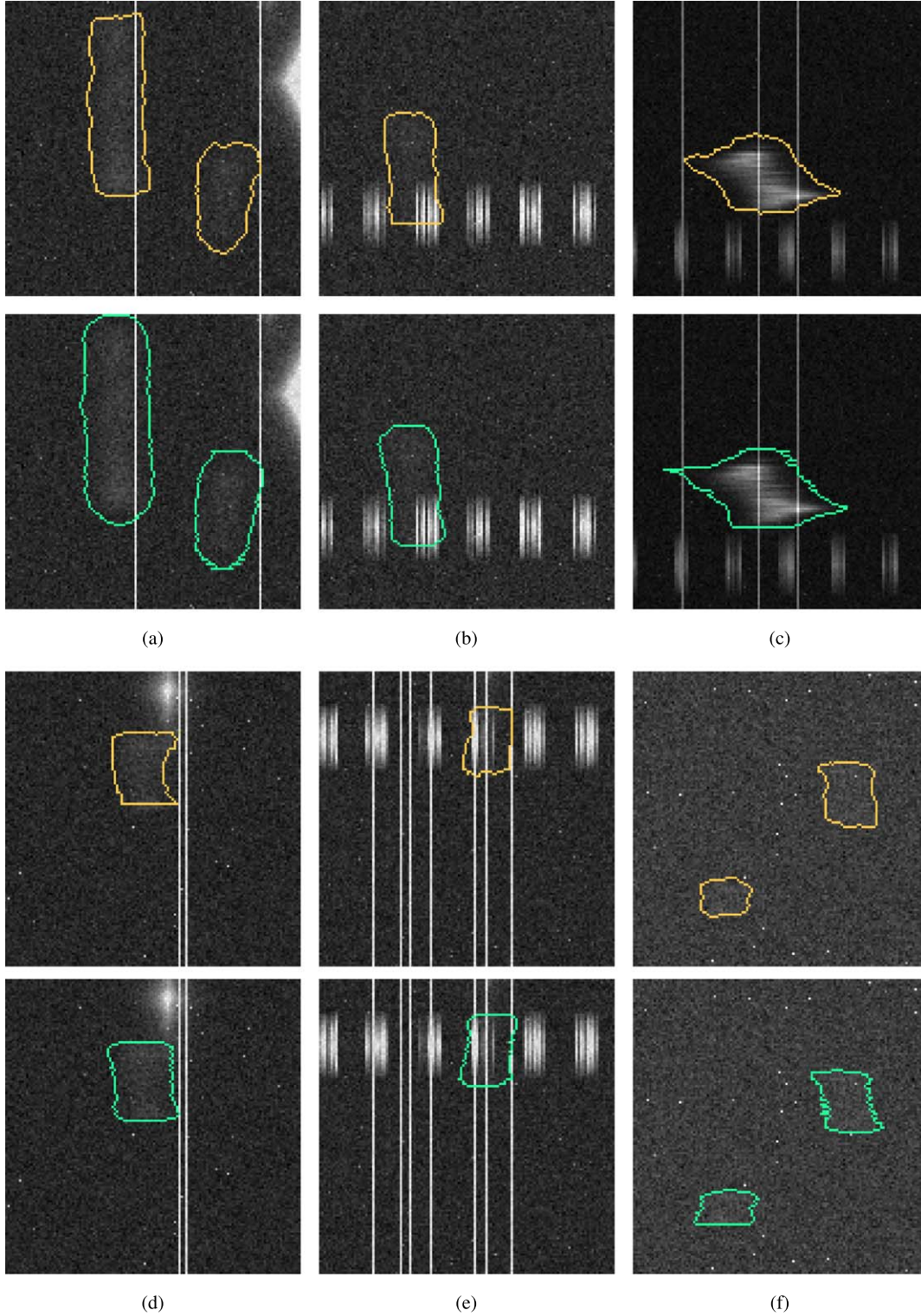


Figure 7. Examples of the H I galaxies correctly recognized by our model. These images are all plotted by part of our final simulated TOD data, with the horizontal axis representing time, the vertical axis representing frequency, and the brightness representing the value of the antenna temperature. In the images, a brighter (whiter) pixel represents a higher temperature at that point, with the brightest areas reaching about 3 K. Yellow lines delineate the galaxy contour determined by the model's output mask. The green lines represent the ground truth, which is the galaxy contour in our simulated TOD data. Other bright areas in the images correspond to various RFI and noise.

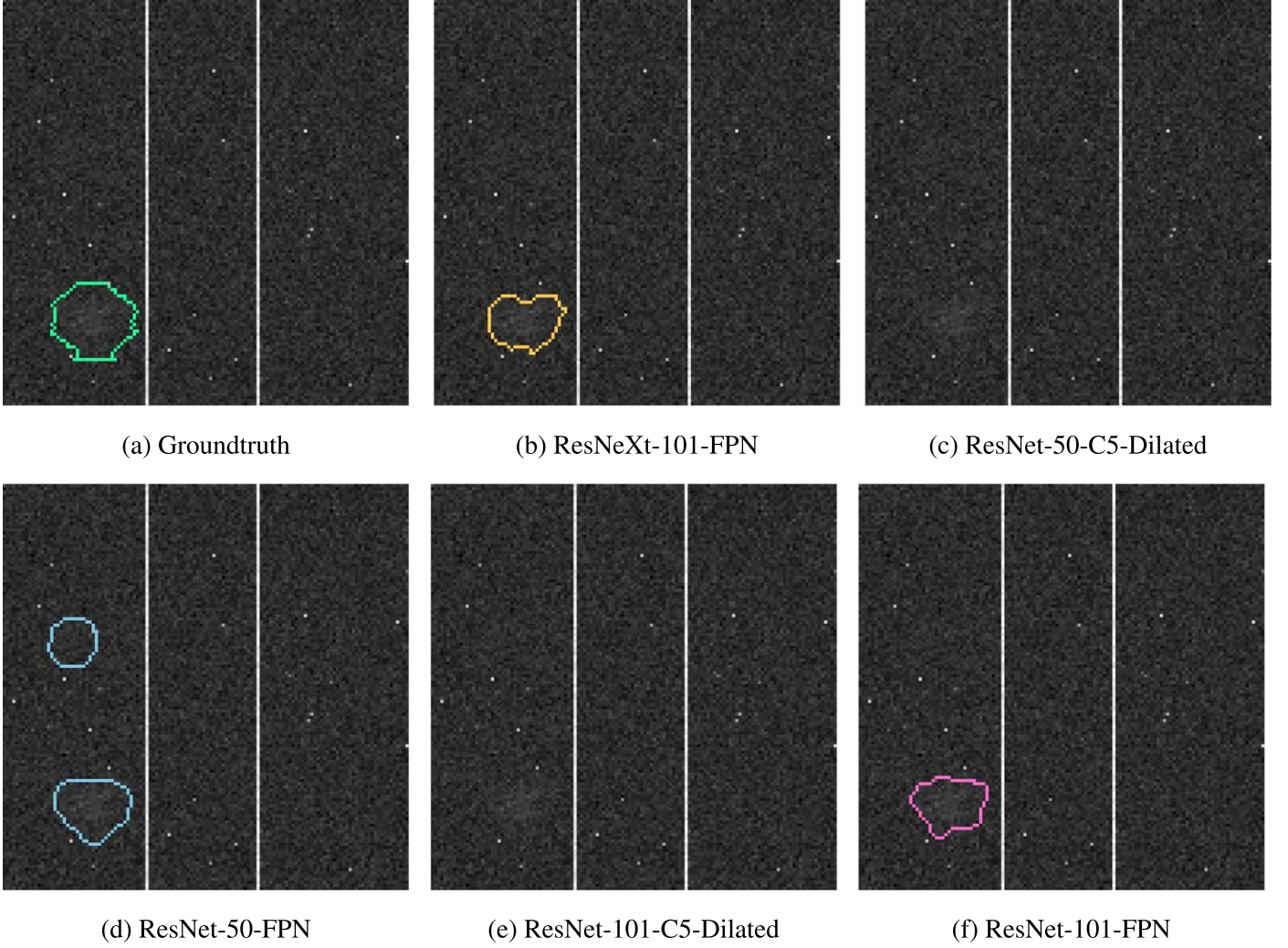


Figure 8. An example from the results of our model with different backbones. The presentation of these images is the same as in Figure 7, and each image represents the recognition results of the model with the corresponding backbone. Specifically, the green lines represent the galaxy contour determined by the ground truth, while the yellow, blue, and pink lines represent the galaxy contour determined by our model with ResNeXt-101-FPN, ResNet-50-FPN and ResNet-101-FPN backbones, respectively.

presents a typical situation where a galaxy and an easily confused RFI portion overlap. As can be seen, after sufficient training, the model can effectively handle such cases.

The RFI contamination is more intense in Figure 7(e), affecting almost half of the galaxy's signal, yet the model still recognizes the galaxy, demonstrating its strong identification capability. Figure 7(f) shows that the model can successfully identify faint galaxies with lower signal-to-noise ratios. Both Figures 7(a) and (f) contain more than one H I galaxy target, highlighting the necessity of performing instance segmentation for galaxies.

Considering the diverse morphology of galaxies and the variety of RFI patterns, our model demonstrates strong generalization capabilities, indicating that it can successfully

accomplish the task of instance segmentation for finding H I galaxies in RFI-contaminated data.

We present the different recognition results of the same example using different backbones as well, with the outcomes of various models marked with distinct color contours. Figure 8 shows a rather faint galaxy, with the two ResNet models using C5-Dilated failing to detect the target. In contrast, ResNet-50-FPN produces a false detection, possibly due to the influence of some faint galaxies with a low signal-to-noise ratio in the training data set, leading the model to misinterpret random noise fluctuations as galaxies. Figure 9 displays a galaxy contaminated by broadband RFI, with ResNet-101-C5-Dilated still missing the target, while ResNet-101-FPN produces a false detection. These examples illustrate that, in some cases, galaxy

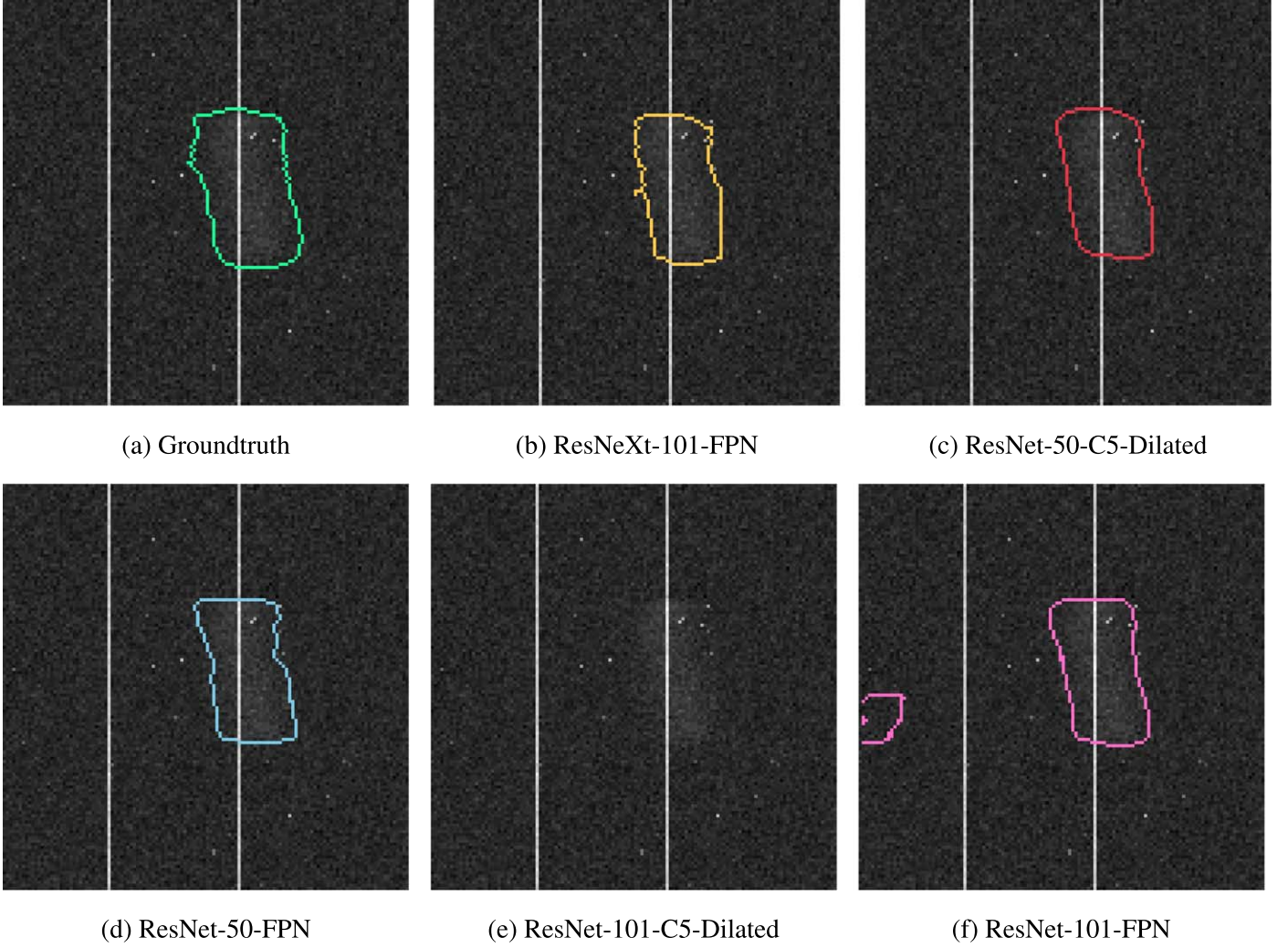


Figure 9. Another example from the results of our model with different backbones. The presentation of these images is the same as in Figure 7, and each image represents the recognition results of the model with the corresponding backbone. The meaning of lines in different colors is the same as in Figure 8, and the red lines represent the galaxy contour determined by our model with ResNet-50-C5-Dilated as the backbone.

recognition can be quite challenging, and models may have limitations and areas for improvement.

6. Discussion

6.1. Construction of the Dataset

In fact, to accomplish the task of finding galaxies through instance segmentation, there are various ways to construct a data set. One of the most direct methods is to label galaxies in the actual telescope observational data and use the label masks as the ground truth. This approach has two execution strategies: one is to use manual labels (similar to the process in many other instance segmentation tasks). Although simple and direct, this requires a certain amount of human labor for labeling and is not easily scalable. And manual labeling is always required

each time when applying this machine-learning method to a new telescope. The second strategy is to use existing methods (e.g., template function method used by ALFALFA) for labeling, but this will cause a certain degree of “distortion” in the ground truth. This is because the recall and accuracy of existing methods are not 100%, which can lead to machine-learning models becoming “similar to existing methods” after training.

Besides using real observational data, another way to construct a data set is to simulate data, such as using simulated galaxy data with real RFI, using real galaxy signals with simulated RFI, or using both simulated galaxy data and simulated RFI with noise background, etc.

In our work, we used simulated galaxy data and simulated RFI. The reason is, first, the differences between our simulated

galaxy signals and the real galaxy signals are minimal, so it is feasible to use simulated galaxy data as a substitute for real galaxy data. Second, it is difficult to search for and label galaxies directly in the TOD data, which is also the reason why we used simulated RFI. It is challenging to separate “pure” RFI without astronomical source signals from real data. Since the galaxy and RFI data are separately simulated, we can accurately and conveniently label them, which greatly assists us in our subsequent work.

It is worth mentioning that the construction of the data set is very flexible. For example, if we want the model to have the ability to identify galaxies among specific RFIs, we can add these particular RFIs (either simulated or real signals) to the original data, allowing the model to “learn” the ability to identify this type of RFI as interference. Furthermore, by adjusting the proportion of faint sources and bright sources in the data set, the model can be more inclined to identify faint or bright source signals. The data set construction method depends on the researcher, but all operations should be performed while ensuring the data is as realistic as possible.

6.2. Model Generalization and Potential Improvements

Although our simulation can obtain observational data for all 19 beams of the FAST telescope for HI galaxies, we only used data from one beam for the final training. This is because, in practice, though the response of different beams to the same signal is actually one of the bases for distinguishing galaxies and RFIs, we have not found a convenient way to simulate the same RFI received by different beams. Using the same RFI data for all beams may cause some errors.

To further train the model, multi-beam data (e.g., FAST’s 19-beam) can be used, inputting different beam data as different channels of two-dimensional data into the network. This allows the model to learn the response information of all beams for the same source and better search for galaxy signals.

In addition, in the real observational data, besides RFI, other influences such as standing waves and bandpass of the system will have an impact on the search results, which means that the real data may be more complex than our simulations and require better detection capability of the model.

Owing to the model’s excellent generalization ability, one can also attempt to apply our network to the detection and extraction of signals from other astronomical sources. In fact, the PointRend Mask R-CNN can effectively perform instance segmentation tasks for numerous categories, while in our work it has only been used for a two-class (galaxies and interference) instance segmentation task. So our network can also be applied to data generated in other stages of telescope data processing for object detection and signal extraction (e.g., Riggi et al. 2023).

Additionally, when training the model, the weights of the different components in the total loss function can be adjusted

to give the model a stronger “inclination.” For instance, increasing the weights of the L_{cls} and L_{box} components in the total loss function can make the model more inclined toward accurate recognition rather than precise segmentation. For a further saying, new network structures can be explored and incorporated into our model to enhance its capabilities for other missions.

It should be noted that, limited by the accuracy of the numerical simulation, we currently consider galaxies with relatively high HI fluxes. Whether our models can perform better for galaxies with lower masses (galaxies fainter than those in our data set) needs to be further investigated. Finally, we will apply our model to real observational data in our subsequent work, trying to perform galaxy searches in real data and comparing the results with other traditional methods.

7. Conclusion

In our work, we constructed a Mask R-CNN network integrated with the PointRend method, aiming to find and extract galaxy signals in radio telescope observational data contaminated by RFI. We simulated the galaxy signals observed by the FAST and the potential RFI impact as realistically as possible, and built a data set based on this simulation for training and testing our network. We compared five different network architectures and chose the best-performing one, ultimately achieving precision and recall of 98.64% and 93.59%, respectively. This demonstrates that our network can successfully accomplish the instance segmentation task of HI galaxy signals in TOD data.

Moreover, thanks to the high-precision detailed performance of the PointRend method, our network can achieve more accurate segmentation when dealing with complex and subtle galaxy structures in astronomical images. We discussed the construction methods of the data set and the possible generalizations and improvements of the model, believing that our network has excellent extensibility and can be applied to other scenarios.

For the extraction of HI galaxy signals, although existing search algorithms have achieved some success in previous projects, there are still some drawbacks and challenges in practical applications. Our bold attempt to find galaxy signals using a deep neural network is an innovative application of machine-learning methods to this task, which helps to provide more reliable basic data for subsequent astronomical analyses and lays a better foundation for the next step of scientific research.

Acknowledgments

We thank Long Xu and Dong Zhao for their useful discussions. We acknowledge the support by the National SKA Program of China, No.2022SKA0110100, and the CAS Interdisciplinary Innovation Team (JCTD-2019-05). We also

acknowledge the science research grants from the China Manned Space Project with No. CMS-CSST-2021-B01.

ORCID iDs

Ruxi Liang  <https://orcid.org/0009-0009-0003-1628>

Furen Deng  <https://orcid.org/0000-0001-8075-0909>

Yichao Li  <https://orcid.org/0000-0003-1962-2013>

References

- Ade, P. A., Aghanim, N., Arnaud, M., et al. 2016, *A&A*, **594**, A13
- Akeret, J., Chang, C., Lucchi, A., & Refregier, A. 2017a, *A&C*, **18**, 35
- Akeret, J., Seehars, S., Chang, C., et al. 2017b, *A&C*, **18**, 8
- Ball, N. M., & Brunner, R. J. 2010, *IJMPD*, **19**, 1049
- Baron, D. 2019, arXiv:1904.07248
- Booth, R. S., & Jonas, J. L. 2012, *AfrSk*, **16**, 101
- Deng, F., Gong, Y., Wang, Y., et al. 2022, *MNRAS*, **515**, 5894
- Dewdney, P. E., Hall, P. J., Schilizzi, R. T., & Lazio, T. J. L. W. 2009, *Proc. IEEE*, **97**, 1482
- Diemer, B., Stevens, A. R., Forbes, J. C., et al. 2018, *ApJS*, **238**, 33
- Forsyth, D. A., & Ponce, J. 2011, *Computer Vision: A Modern Approach* (2nd Edn.; Upper Saddle River, NJ: Prentice Hall)
- Fridman, P. A., & Baan, W. A. 2001, *A&A*, **378**, 327
- Giovanelli, R., & Haynes, M. P. 2015, *A&ARv*, **24**, 1
- Giovanelli, R., Haynes, M. P., Kent, B. R., et al. 2005, *AJ*, **130**, 2598
- Gnedin, N. Y., & Kravtsov, A. V. 2011, *ApJ*, **728**, 88
- Goodfellow, I., Bengio, Y., & Courville, A. 2016, *Deep Learning* (Cambridge, MA: MIT Press)
- Goyal, P., Dollár, P., Girshick, R., et al. 2018, arXiv:1706.02677
- He, K., Gkioxari, G., Dollár, P., & Girshick, R. 2017, in *Proc. of the IEEE Int. Conf. on Computer Vision (ICCV)* (Venice: IEEE), 2961
- He, K., Zhang, X., Ren, S., & Sun, J. 2016, in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (Las Vegas, NV: Proc. IEEE), 770
- Jiang, P., Tang, N.-Y., Hou, L.-G., et al. 2020, *RAA*, **20**, 064
- Johnston, S., Taylor, R., Bailes, M., et al. 2008, *ExA*, **22**, 151
- Jones, M. G., Haynes, M. P., Giovanelli, R., & Moorman, C. 2018, *MNRAS*, **477**, 2
- Kerrigan, J., Plante, P. L., Kohn, S., et al. 2019, *MNRAS*, **488**, 2605
- Kilborn, V. A. 2001, PhD thesis: The Large-Scale Distribution of Neutral Hydrogen in the Local Universe, The University of Melbourne
- Kirillov, A., Wu, Y., He, K., & Girshick, R. 2020, in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)* (Seattle, WA: IEEE), 9796
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. 2012, *ImageNet Classification with Deep Convolutional Neural Networks in Advances in Neural Information Processing Systems (NIPS)*, Vol. 25 (Lake Tahoe, CA: NIPS)
- Lin, T.-Y., Dollar, P., Girshick, R., et al. 2017, in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (Honolulu, HI: IEEE), 2117
- Meyer, M. J., Zwaan, M. A., Webster, R. L., et al. 2004, *Mon. Not. R. Astron. Soc.*, **350**, 1195
- Nan, R., Li, D., Jin, C., et al. 2011, *Int. J. Modern Phys. D*, **20**, 989
- Offringa, A. R., de Bruyn, A. G., Biehl, M., et al. 2010, *MNRAS*, **405**, 155
- Pillepich, A., Springel, V., Nelson, D., et al. 2018, *MNRAS*, **473**, 4077
- Pinchuk, P., & Margot, J.-L. 2022, *AJ*, **163**, 76
- Qian, N. 1999, *Neural Netw.*, **12**, 145
- Ren, S., He, K., Girshick, R., & Sun, J. 2015, in *Advances in Neural Information Processing Systems (NIPS)*, Vol. 28 (Montréal: NIPS)
- Riggi, S., Magro, D., Sortino, R., et al. 2023, *A&C*, **42**, 100682
- Saintonge, A. 2007, *AJ*, **133**, 2087
- Sun, H., Deng, H., Wang, F., et al. 2022, *MNRAS*, **512**, 2025
- Sutskever, I., Martens, J., Dahl, G. E., & Hinton, G. E. 2013, *PMLR*, **28**, 1139
- Tolley, E., Korber, D., Galan, A., et al. 2022, *A&C*, **41**, 100631
- Weinberger, R., Springel, V., Hernquist, L., et al. 2017, *MNRAS*, **465**, 3291
- Xie, S., Girshick, R., Dollar, P., Tu, Z., & He, K. 2017, in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (Honolulu, HI: IEEE), 1492
- Yang, Z., Yu, C., Xiao, J., & Zhang, B. 2020, *MNRAS*, **492**, 1421
- Yu, F., & Koltun, V. 2016, arXiv:1511.07122